REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
7/27/2006

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
10 Mar 2003- 28 Feb 2006

4. TITLE AND SUBTITLE
Cartesian Grid Methods for Moving Geometries

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-03-1-0122

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Marsha J. Berger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
New York University
251 Mercer St.
New York, NY 10012

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
4015 Wilson Blvd
Mail Room 713
Arlington, VA 22203

Dr Fariba Fahroo

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

AFRL-SR-AR-TR-06-0429

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Many classes of physical problem can be models through the use of sets of linear equations. The solution of the sets of equations is equivalent to calculation of a matrix inverse or generalized inverse, or to the reduction of the matrix to some type of canonical form, including determination of characteristic equation. Conventional machine computation relies on p-ary (for a radix number p such as 2 or 10), or floating-point computation, poor conditioning in connection with round-off error can result in unreliable answers. For scientific computations related to quantum physics, a possible approach is to use techniques of exact linear computation

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU		Chao Lu
					19b. TELEPHONE NUMBER *(Include area code)* 410404-3701

Final Report
Cartesian grids for Moving Geometries
AFOSR F49620-03-1-0122

Marsha J. Berger
Courant Institute
251 Mercer St.
New York, NY 10012

Abstract

Our long term goal has been the development of automated tools for computing fluid flow in complicated geometries. With the help of long range funding, we have developed mesh generation tools, surface preparation algorithms, and a highly scalable parallel inviscid steady state flow solver using Cartesian grids with embedded boundaries. This has enabled us to do high fidelity inviscid engineering calculations for a wide variety of complex configurations. The goal of our most recent work is to develop methods to simulate time-dependent flows for bodies in relative motion. This capitalizes on the robustness and ease of Cartesian grid generation, since for bodies in relative motion a new grid must be generated after each step. This research is in collaboration with Michael Aftosmis and Scott Murman at NASA Ames Research Center. Technology transfer is facilitated by our Cart3D code, which is used by over 100 groups around the country.

Introduction

Cartesian grids have proven themselves in the last decade for inviscid flow simulations around complex geometries. There are many flow situations for which their rapid turnaround time and level of automation have had great success. Cartesian grid methods have been able to incorporate very complicated geometries quickly without user intervention [2]. High resolution finite volume flow solvers have been developed that can accurately discretize and robustly solve

on Cartesian meshes with embedded boundaries [1]. Their simplicity and data locality on these meshes lead to high performance on large parallel processors [3,4].

The goal of our recent research was to develops algorithms for accurate and robust flow solutions for geometries in relative motion. The interesting mathematical challenges lie in handling the irregular cells where the geometry intersects the grid. These problems are surmountable when the geometry does not move relative to the mesh, but new issues arise in treating moving bodies. The mesh itself does not move, so there is no issue of mesh smoothness or tangling. The new difficulties with embedded boundaries have to do with the computational cells that are exposed or covered during a time step as the body moves through the mesh. There were two aspects to this work. First, the steady-state solver was converted to a time-dependent solver using pseudo-time integration. In this approach, an artificial time derivative, call it $\partial u/\partial\tau$ is added to an implicit time-discretization of the Euler equation, and the so-called psuedo-time is marched to steady state. This leverages all the machinery of the steady-state flow solver. As part of this effort, we needed to improve the convergence of the latter, which was felt was being hampered by the limiters in the finite volume scheme. This led to a detailed investigation of the use of limiters on irregular grids. Second, we started investigating finite volume methods for moving geometry. Both efforts are summarized below.

Limiters for Finite Volume Schemes

The Cart3D steady state flow solver has some stalling of convergence due to cut cells. This became critically important in the time-dependent case, since the solution to the implicit discretization needs to converge at every step Much of the problem was due tot he behavior of the limiters on irregular grids, which includes mesh refinement interfaces with 2:1 mesh ratios as well as irregular cells adjacent to a body.

We have developed a theory in one dimension for non-uniform meshes, and extended several limiters in common use today to the irregular case. Most limiters use the Sweby formulation, where limiters are written as a function of $R = \frac{u_{i+1}-u_i}{u_i-u_{i-1}}$, for example the van Leer limiter has $\phi(R) = \frac{4R}{(1+R)^2}$. To preserve second-order accuracy, which treats a linear solution exactly, a limiter should not limit this case. A linear solution thus has $R = 1$, and $\phi(R) = 1$.

However on irregular grids, $R \neq 1$ for a linear solution. This can be easily remedied by incorporating the mesh widths into the definition of R, defining $R' = \frac{(u_{i+1}-u_i)/h_{i+1/2}}{(u_i-u_{i-1})/h_{i-1/2}}$ so that instead of undivided differences, we use divided

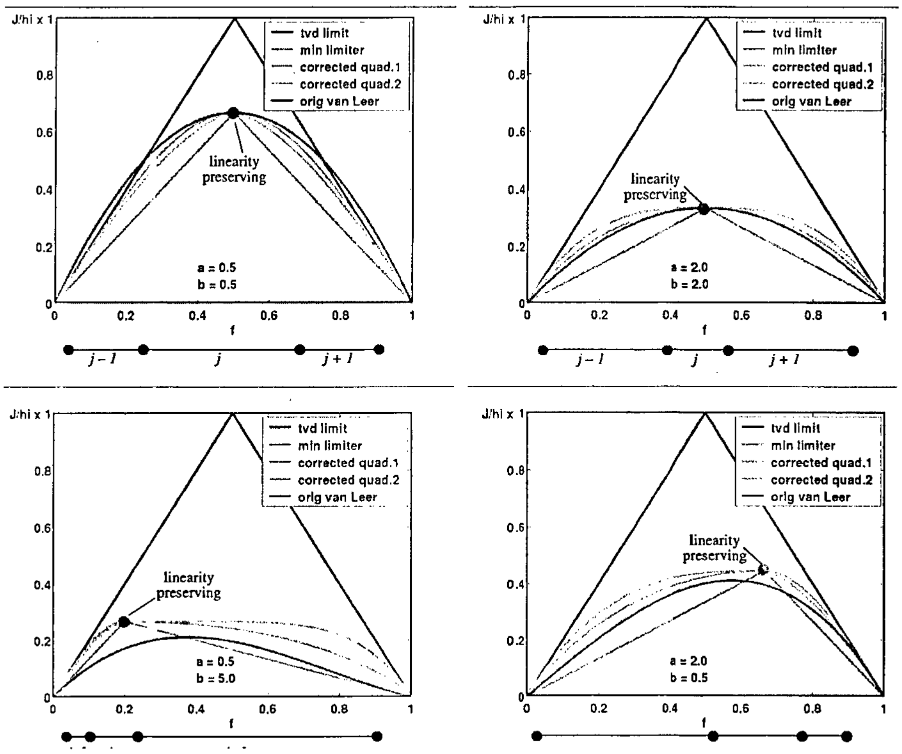

Figure 1: Limiters evaluated on stretched meshes, with irregular mesh cells sketched at the bottom

differences. This will maintain $\phi(R' = 1) = 1$, but unfortunately it is not TVD for the rest of the interval.

By analyzing and generalizing the form of the limiters, we have devised new functional forms that maintain monotonicity for all mesh ratios. Figure 1 shows several cases with different mesh ratios, the new limiters we have developed that generalize van Leer, and the original van Leer limiter that is not TVD. This work is described in [4].

Work on multi-dimensional limiters in still in progress. Since cell centroids are not coordinate-aligned, the work above is not sufficient to handle this case. We are pursuing two avenues. One generalizes the one-dimensional work by replacing one sided differences with reconstructed least-squares derivatives. This is linearity preserving, but not sufficiently monotone, since it admits an odd-even mode. However in practise it is the limiter of choice for cut cells at this point. Another approach we are investigating uses linear programming to limit the gradient according to constraints on neighbors, while preserving as much as possible of the originally reconstructed gradient. This appears to be very

accurate, but the next step is to develop an efficient solution technique, and test it with regards to limiter chatter. This work will continue over the next grant period.

Relative Motion in Finite Volume Schemes

We are using a two-pronged approach to develop a relative motion simulation tool. We are using both a one dimensional model problem, and a three-dimensional "baseline" problem that consists of simply putting together the individual pieces of the algorithm in a static, non-parallel way, using the recently developed time-dependent (but with static geometry) flow solver and a 6-DOF module for rigid body motion written for Cart3D by my collaborator Scott Murman.

One-dimensional Model Problem

The simplest setting to study the accuracy and stability of flow with a moving wall is in one space dimension. We use both linear advection and a non-linear piston modeled with the Euler equations. The wall can be moving either into or away from the computational domain. Consider the equation $u_t + f(u)_x = 0$, and integrate over a slab in space as in figure 2, with the left side of the slab located at $x_0 + wt$, and the right side fixed at x_1, where w is the speed of the wall. Using the Leibniz rule and then the equation, we get

$$\begin{aligned}
\frac{d}{dt}\int_{x_0+wt}^{x_1} u(x,t)\,dx &= \int_{x_0+wt}^{x_1} u_t\,dx \; - w \cdot u(x_0+wt,t) \\
&= \int_{x_0+wt}^{x_1} -f_x\,dx \; - w \cdot u(x_0+wt,t) \qquad (1) \\
&= -f(u(x_1,t)) + f(u(x_0+wt,t)) \; - w \cdot u(x_0+wt,t)
\end{aligned}$$

This is the basis of the numerical discretizations. We are primarily interested in implicit finite volume discretizations, to avoid stability problems at the cut-cells as well as to allow taking larger time steps than an explicit scheme would allow. Implicit schemes are common in aerodynamics due to the variety of scales, not all of which need to be resolved.

For linear advection $u_t + a\,u_x = 0$ (taking $f(u) = au$) the wall speed w determines whether the wall should be an inflow or outflow boundary. We have developed a methodology to study the stability of discretizations of this equation using the initial-boundary value stability theory of GKS ("Stability Theory of

Difference Approximations for Mixed Initial Boundary Value Problems, II". Math. Comp. 26, 619-686, (1972)).

The stability results obtained so far use the first-order upwind finite volume scheme, to make the algebra tractable. We can show results such as:

> Prop.: For the implicit upwind finite volume scheme applied to a right-moving wall with any CFL, if the interior scheme is stable (which it is), then the moving wall scheme is stable.

In other words, allowing an arbitrarily small cell volume αh for the cut cell at the wall, with $\alpha \approx 0$, does not introduce a numerical stability problem. For a wall moving away from the domain we study two approaches to computing the cut cell update: cell merging, or the more exact "crossing time" calculation, where the exact time that the wall crosses cell edges is used in the update formulas. Not surprisingly, both of those are stable as well.

With this background preparation, we are trying to develop a free-stream preserving one-dimensional method using a splitting paradigm. Unlike the more usual splitting methods that split an equation by spatial derivatives, we want to split the terms involving geometry motion from those involving an update of the solution in time. We want a scheme where at time t_n

1. the geometry moves *instantaneously* to its new position,

2. the flow is updated using only the geometry in the new location.

This is illustrated by the approximate geometry motion in figure 3b, which we also call the backward Euler approximation to the geometry.

A first order splitting method along these lines first moves the geometry,

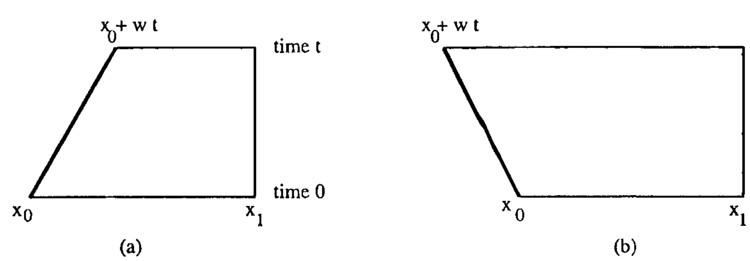

Figure 2: Notation for moving wall equation (1). In (a) the wall is on the left moving with speed w into the computational domain, where $w > 0$; in (b) it is moving away with speed $w < 0$.

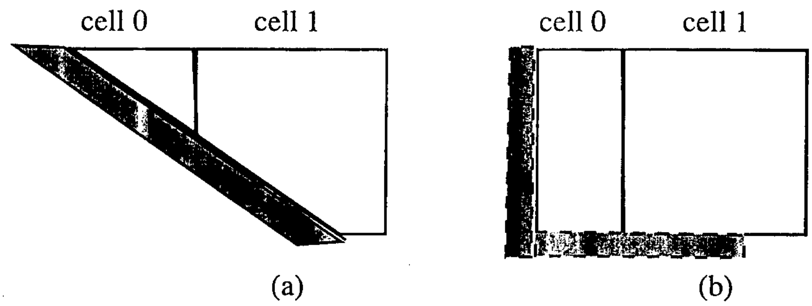

Figure 3: (a) Illustration of cells 0 and 1 with left-moving wall. (b) The "backward Euler" approximation to the geometry (use the geometry at time $n+1$ and include the terms that make it free-stream preserving (i.e. so that the cell closes).

giving the intermediate solution denoted by bars

$$\begin{aligned} \alpha h\,\overline{u_0} &= 0 + u_b\,\Delta V_0 \\ h\overline{u_1} &= \alpha h\,u_0^n + u_b\,\Delta V_1 \end{aligned} \tag{2}$$

where ΔV_j is the change in volume between time $n+1$ and n for cell j. The solution is then updated on fixed geometry to get

$$\begin{aligned} \alpha h\,u_0^{n+1} &= \alpha h\,\overline{u_0} - a\Delta t(u_0^{n+1} - u_b) \\ hu_1^{n+1} &= \quad h\overline{u_1} - a\Delta t(u_1^{n+1} - u_0^{n+1}) \end{aligned} \tag{3}$$

This is actually the usual first-order free-stream-preserving method written in two-step form, to make it easier to think of generalizations. One such generalization gives a second-order splitting method along these same lines, using linear interpolation to determine the intermediate values of the cut cells is:

$$\begin{aligned} \overline{u_1} &= LinearExtrap\,(u_b^n, u_2^n) \\ \overline{u_0} &= LinearExtrap\,(u_b^n, u_2^n) \\ \overline{u_b} &= LinearExtrap\,(u_b^n, u_2^n) \end{aligned} \tag{4}$$

and

$$\begin{aligned} u_1^{n+1} &= \overline{u_1} - \frac{a\Delta t}{2h}[(u_{3/2}^{n+1} + \overline{u_{3/2}^n}) - (u_{1/2}^{n+1} + \overline{u_{1/2}^n})] \\ \alpha u_0^{n+1} &= \alpha\overline{u_0} - \frac{a\Delta t}{2h}[(u_{1/2}^{n+1} + \overline{u_{1/2}^n}) - (u_b^{n+1} + \overline{u_b})]. \end{aligned} \tag{5}$$

In [7], the interpolation uses the specified boundary condition and the solution in the first full flow cell. (Using the solution in cells closer to the moving wall is not stable).

Numerical experiments shown in table 1 clearly show second order convergence for our linear advection test problem with a cosine initial condition, with a CFL of $\lambda = 2.0$, a small cell ratio $\alpha = .1$, and the wall moving one cell per time step. The errors are quite comparable to a second-order unsplit scheme (not shown here). A manuscript describing this work is in preparation.

Unfortunately the linear problem is really too easy to serve as a good model for the multi-dimensional nonlinear problem. Even a one-dimensional moving piston is much more complicated than the linear model. The moving wall is a characteristic boundary, and there is only one boundary condition associated with it. Without a specified inflow boundary condition to help interpolate the solution onto a changing mesh, it is very difficult to find both a stable and second-order accurate splitting method for systems of equations. Here we show some experiments using an unsplit method. Figure 4 show results of a piston moving right, so that the compression side is on the right, and a rarefaction on the left. The experiments use either Backward Euler or the trapezoid rule in time and various first and second-order methods in space, as detailed in the caption. The piston moves 3 cells per time step. Note that although it is stable, the trapezoid rule is not monotone for this CFL number. A related issue that will need investigation is monotonicity of limiters for implicit methods.

The 1D moving piston is a good model problem in that it exercises the numerical methods with the relevant physics of compression and rarefaction waves. However in some ways it is actually a harder problem than we will find in higher dimensions, since the flow is determined entirely by the piston motion.

# Cells	2 norm error	max. norm error
25	4.19E-02	7.30E-02
50	8.52E-03	1.52E-02
100	2.02E-03	3.63E-03
200	4.95E-04	8.90E-04
400	1.22E-04	2.20E-04

Table 1: Second order split method, where geometry is moved instantaneously, then the solution is updated using a second order method on fixed geometry. The uncovered cells were initialized by linear extrapolation using the boundary value at t_n and u_2.

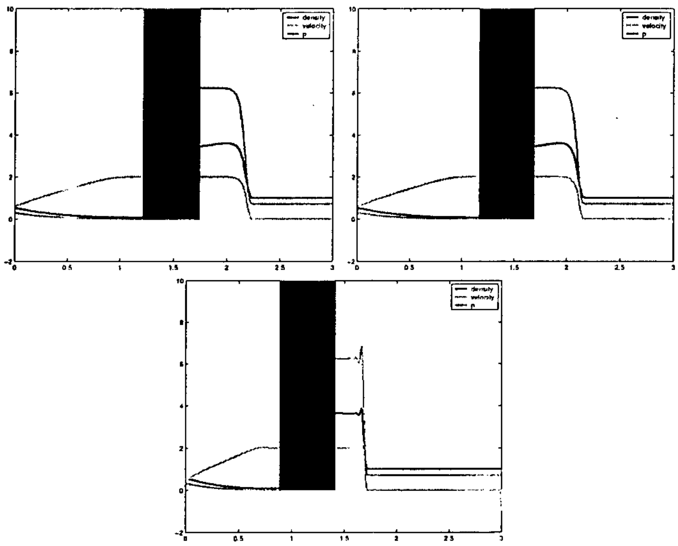

Figure 4: Piston moving to the right. The computations from left to right use backward Euler in time with no gradients, backward Euler with limited gradients, and the trapezoid rule with limited gradients. The piston moves 3 cells per time step. Note that the implicit trapezoid rule is not monotone at this cfl. Blue shows density, green shows velocity and purple is pressure.

In a store separation problem for example the flow is determined mostly by the stationary geometry. Also, errors show up much more strongly in one dimension, since there is no place for the error to hide.

Three-dimensional Experiments

Although we do not yet have the algorithm of choice in hand, we have implemented a first pass algorithm in three space dimensions by putting together the pieces of the static algorithm. For example, at every time step a new mesh was generated, even though only a small fraction of the domain needed to be re-meshed due to the moving geometry. The steady-state code was extended to the time-dependent case with an implicit discretization in time, and a dual-timestepping formulation that iterates until convergence at each time step. The spatial discretization, multigrid and parallelization of the steady-state code were re-used as is. The cut-cell discretizations used the first-order backward Euler in time [5]. This work was not conservative unless the CFL relative to the moving geometry was limited to less than 1, which is much too restrictive in many

cases. Nevertheless we have been able to run several cases of interest [6,7]. This experience also tells us where we will need to develop new infrastructure.

More recently we have done simulations of a PPB (practice plastic bomb) separating from a TER-9 ejection rack mounted on an F-16, shown in Figure 5. A zoom on the PPB is shown in figure 6. There is experimental data from Eglin that we used for verification. Our results to date show that the displacements can be well approximated, but the moments are extremely difficult and we do not do a good job. One hypothesis is that this is due to lack of conservation around the separating store. (We point out however that the Seek Eagle simulations using Beggar (J.M. Lee, K. Dunworth, B. Chesser, S. Ellison and B. Jolly. "CFD Investigation of Plastic Practice Bomb (PPB) Separation from F16/TER-9A Configuration". AIAA Paper 2003-4224, June, 2003.) also had difficulty with this.) This regime, where the flow is almost supersonic, and the PPB weighs only 8 pounds, is extremely difficult to compute.

Publication

[1] M. Aftosmis, M. Berger, and G. Adomavicius."A Parallel Multilevel Method for Adaptive Refined Cartesian grids with Embedded Boundaries". AIAA Paper 2000-0808, Jan. 2000.

[2] M.Aftosmis, M. Berger, and J. Melton. "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry". AIAA Journal V. 36 (6), June, 1998.

[3] M.Aftosmis, M. Berger, and S. Murman. "Applications of Space-Filling-Curves to Cartesian Methods for CFD". AIAA Paper 2004-1232, Jan., 2004.

[4] M. Berger, M. Aftosmis, and S. Murman. "Analysis of Slope Limiters on Irregular Grids". AIAA Paper 2005-0490, Jan., 2005.

[5] S. Murman, M. Aftosmis and M. Berger. "Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method", AIAA Paper 2003-1119, Jan. 2003.

[6] S. Murman, M. Aftosmis and M. Berger. "Simulations of Store Separation from an F/A-18 with a Cartesian Method", AIAA Journal of Aircraft, Vol. 41 No. 4, 2004.

[7] S. Murman, M. Aftosmis and M. Berger. "Numerical Simulation of Rolling-Airframes using a Multi-Level Cartesian Method", AIAA Journal of Spacecraft and Rockets, 41(3), 2004.

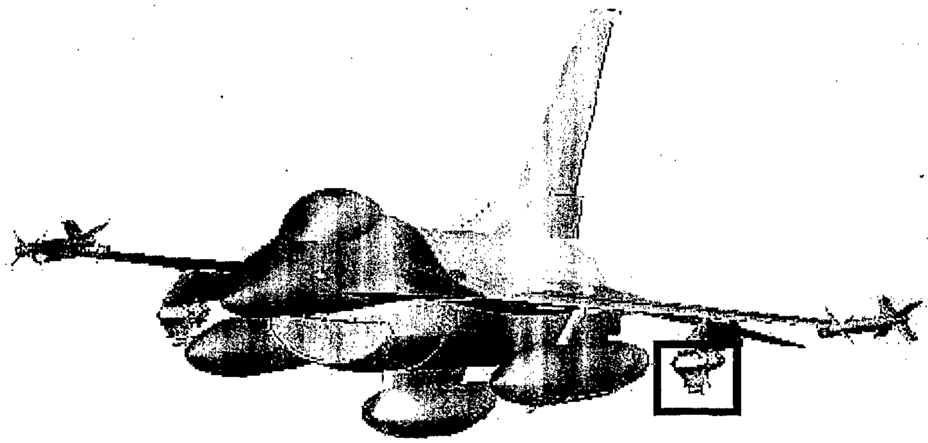

Figure 5: F-16 including sidewinders and fuel tank. The TER-9 ejection rack and PPB is in the rectangle.

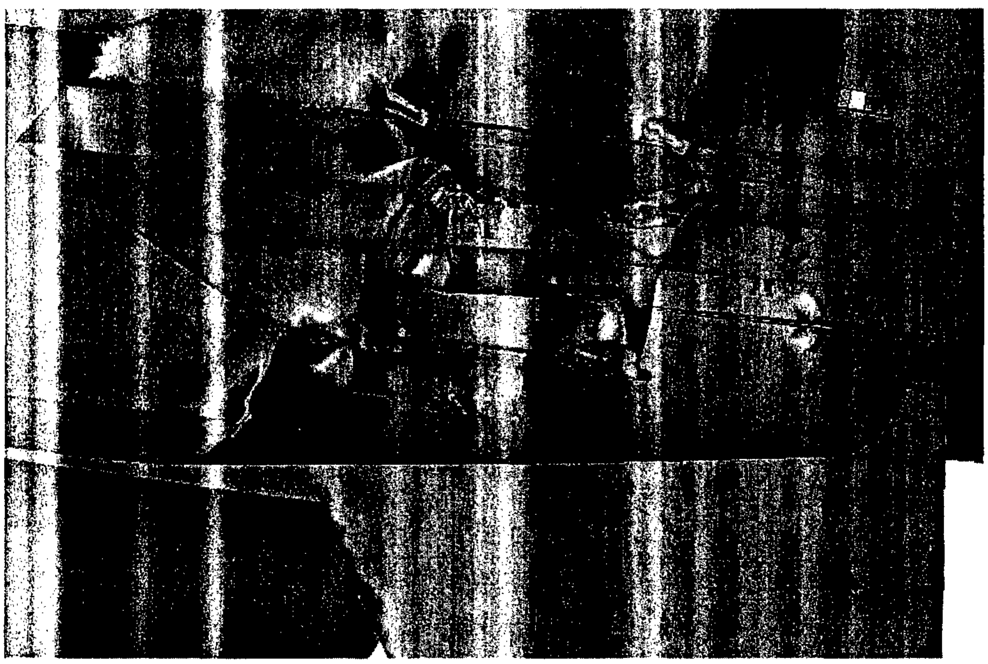

Figure 6: Zoom of PPB separating from an F-16, colored by surface pressures.